

## PEMBUATAN GAME TOWER DEFENSE MENGGUNAKAN

### ALGORITMA A\*

**Aloysius Wiranata**

Teknik Informatika / Fakultas Teknik Universitas Surabaya  
[s6134030@student.ubaya.ac.id](mailto:s6134030@student.ubaya.ac.id)

**Budi Hartanto**

Teknik Informatika / Fakultas Teknik Universitas Surabaya  
[budi@staff.ubaya.ac.id](mailto:budi@staff.ubaya.ac.id)

**Marcellinus Ferdinand Suciadi**

Teknik Informatika / Fakultas Teknik Universitas Surabaya  
[ferdi@staff.ubaya.ac.id](mailto:ferdi@staff.ubaya.ac.id)

**Abstrak** – Permainan tower defense yang sudah ada sekarang ini kurang memberikan pemain kebebasan dalam bermain dimana pemain tidak dapat membuat jalurnya sendiri. Oleh sebab itu, dibuatlah sebuah permainan tower defense berbasis desktop yang memberikan pemain kebebasan dalam menentukan jalur monster. Permainan ini dibuat menggunakan unity dan bersifat single player. Pada permainan tower defense ini, terdapat sebuah objektif yang harus dilindungi oleh pemain. Nantinya, monster yang muncul akan berusaha mencari jalannya sendiri menuju objektif tersebut. Algoritma pencarian jalan yang digunakan oleh monster tersebut adalah algoritma A\*. Algoritma A\* ini merupakan algoritma pathfinding yang paling sering digunakan dalam berbagai jenis game. Algoritma A\* pada tugas akhir ini menggunakan library yang ditulis oleh Aron Granberg. Pemilihan library ini didasari pada dokumentasi yang jelas dan kemudahan dalam menggunakan library tersebut. Selain itu, berdasarkan hasil analisa maka pada permainan ini juga ditambahkan fitur Custom Tower. Custom Tower adalah sebuah fitur dimana pemain dapat memodifikasi atribut dari tower yang ada sesuai dengan keinginannya sendiri. Agar permainan tetap menjadi seru dan tidak ada tower yang terlalu kuat, maka semakin tinggi atribut custom tower, maka biaya pembuatan tower juga semakin tinggi. Dari hasil uji coba permainan didapatkan hasil bahwa dengan adanya custom tower dan custom track mampu meningkatkan keseruan permainan tower defense. Saran yang dapat diberikan untuk pengembangan selanjutnya adalah menambahkan fitur multiplayer dan varian monster dan tower.

**Kata kunci** : Tower Defense, Algoritma A\*, berbasis desktop, single player.

**Abstract** – Nowadays, tower defense games doesn't give its players freedom which they cannot create their own path, therefore a desktop based game of tower defense game which able to give its player freedom in creating their own path is created. This game is built using unity and can be played in single player. In this game, there is an objective that must be protected by its player. There will also be monsters that will try to find their own path to the objective. This monster will be using A\* path finding algorithm. A\* path finding algorithm is the most common path finding algorithm used in many games. This final project will be using A\*

algorithm library implemented by Aron Granberg. The using of this library is based on its clear documentation and easiness of using this library. From the result of analysis, the game will also have feature to create Custom Tower. Custom Tower is a feature that enables its player to modify attributes of a tower according to his own needs. In order to remain exciting and no tower is too overpowered, the higher the attribute of a custom tower then the cost of building that tower will also be higher. From the game test results obtained that the presence of custom tower and custom track increases the enjoyment of playing tower defense game. Suggestions that can be given for further development is to add more variety of monsters and tower and multiplayer feature.

**Keywords:** Tower Defense, A\* Algorithm, Desktop based, single player.

## **1. LATAR BELAKANG**

Permainan tower defense yang sudah ada sekarang ini kurang memberikan pemain kebebasan dalam bermain dimana pemain tidak dapat membuat jalurnya sendiri. Oleh sebab itu, permainan tower defense yang akan dibuat memberikan pemain kebebasan dalam menentukan jalur monster. Pada permainan ini, pemain akan menghadang monster yang berjalan menuju obyektif yang harus dilindungi menggunakan tower. Monster yang berjalan akan mencari jalannya sendiri menuju obyektif, sehingga lokasi tower akan mempengaruhi jalan yang akan ditempuh monster. Obyektif yang digunakan pada permainan ini adalah sebuah tempat penampungan manusia sedangkan untuk monster akan direpresentasikan dalam bentuk virus. Pemain akan dinyatakan kalah apabila gagal melindungi tempat penampungan tersebut. Sedangkan jika pemain berhasil bertahan hingga wave terakhir, maka pemain akan dinyatakan menang.

Monster yang mencari jalannya sendiri menuju obyektif tersebut akan menggunakan algoritma A\* *pathfinding* agar dapat sampai ke tujuannya. Algoritma A\* ini merupakan algoritma pathfinding yang paling sering digunakan dalam berbagai jenis game. Algoritma A\* pada tugas akhir ini menggunakan

library yang ditulis oleh Aron Granberg. Pemilihan library ini didasari pada dokumentasi yang jelas dan kemudahan dalam menggunakan library tersebut. Rumusan masalah pada tugas akhir ini adalah: Bagaimana membuat permainan bergenre tower defense menjadi lebih menarik dimainkan? Tujuan yang hendak dicapai dari pembuatan Tugas Akhir ini adalah membuat permainan tower defense berbasis desktop yang mampu menjadi lebih menarik untuk dimainkan.

## **2. DASAR TEORI**

Dalam pembuatan tugas akhir ini menggunakan tiga teori pendukung yaitu Unity, Algoritma Pathfinding dan Tower Defense.

- Permainan bergenre Tower Defense adalah permainan berjenis strategi yang cukup populer. Hal ini dibuktikan dengan banyaknya jumlah unduhan game berjenis Tower Defense di google *appstore* yang mencapai angka 10 juta. Tujuan dari permainan Tower Defense ialah untuk berlindung dari serangan musuh yang berjalan melalui jalur tertentu yang telah ditentukan. Tower bertujuan untuk menyerang musuh secara terus menerus pada range yang telah ditentukan.
- Unity adalah sebuah alat yang digunakan untuk membuat atau mengembangkan sebuah game. Unity merupakan Game Engine yang bersifat cross-platform. Unity dibuat oleh Unity Technologies sebagai alat pengembang game pada berbagai platform seperti PC, console, website dan mobile device. Unity menggunakan bahasa C# dan JavaScript sebagai bahasa pemrograman untuk script-scriptnya.

Algoritma Pathfinding ialah algoritma yang berfungsi untuk menemukan jalur yang paling tepat mulai dari *vertex* A sebagai vertex awal ke vertex B sebagai vertex akhir. Contoh algoritma pathfinding adalah algoritma *Breadth-First Search* (BFS), Dijkstra, Greedy BFS dan A\*. Menurut Dalmau (2004:228) A\* adalah salah satu algoritma *pathfinding* yang banyak digunakan untuk membuat *game*, ia juga mengatakan bahwa algoritma A\* telah banyak digunakan dalam game yang bersifat real-time. Sedangkan menurut Patel (2014), Algoritma A\* menggabungkan kedua kekuatan algoritma Dijkstra dan Greedy Best First search sehingga dapat dikatakan algoritma A\* mampu menemukan jalur terpendek dalam waktu yang tercepat bila dibandingkan dengan algoritma Dijkstra.

### **3. ANALISA**

Hasil analisa diperoleh dari proses benchmarking pada aplikasi yang memiliki genre sejenis. Aplikasi permainan yang digunakan sebagai pembanding antara lain : Plant vs Zombie, Kingdom Rush, dan Warcraft III : Element Tower Defense. Setelah menganalisis keadaan yang ada saat ini, ditemukan beberapa permasalahan, yaitu :

- Banyak game tower defense yang belum menyediakan '*Custom Track*' bagi pemainnya sehingga permainan terkesan datar dan kurang menantang. Dengan adanya '*Custom Track*' pada permainan tower defense akan membuat permainan dengan genre strategi ini lebih memerlukan keahlian dalam berpikir dan merancang strategi.

- Pada beberapa sistem permainan saat ini tidak memberikan indikator jarak jangkauan serangan tower sehingga pemain tidak dapat mengetahui area mana yang dapat dijangkau tower, dan tidak dapat dijangkau tower.
- Permainan dapat menjadi membosankan apabila jumlah tower atau monster atau wave kurang bervariasi.

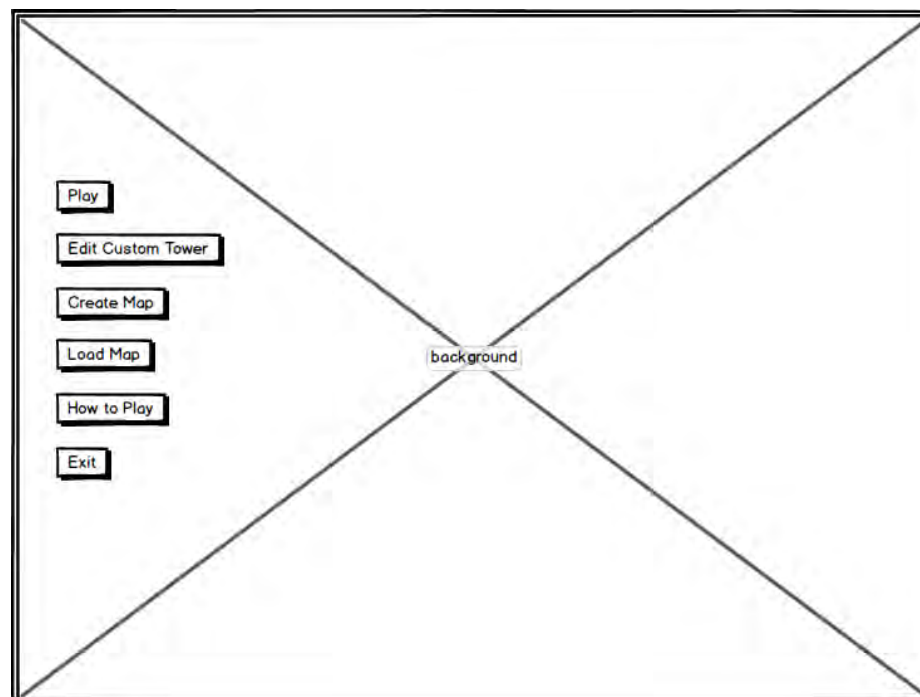
Dari analisis masalah tersebut maka akan terdapat beberapa fitur-fitur pada sistem yang akan dibuat. Fitur tersebut digolongkan menjadi fitur kebutuhan umum dan kebutuhan khusus. Dimana fitur kebutuhan umum meliputi highscore, upgrade tower, peletakkan tower yang bebas, indikator jarak serangan dan bonus skor bagi pemain yang dapat membunuh monster dengan cepat. Sedangkan untuk kebutuhan umum meliputi Custom track yang dapat dibuat sesuai dengan keinginan pemain.

#### **4. DESAIN**

Desain yang akan dibuat antara lain desain permainan, desain proses, desain pencarian jalan, desain data dan desain tampilan (User Interface). Untuk desain pencarian jalan, akan digunakan Library algoritma A\* yang dapat diunduh dari [http://arongranberg.com/astar/downloads/0/master\\_Free](http://arongranberg.com/astar/downloads/0/master_Free). Pemilihan penggunaan library ini didasarkan pada kelebihan-kelebihan yang dimiliki oleh library ini. Beberapa kelebihan tersebut adalah kemudahan implementasi dari library ini dan kejelasan dokumentasi yang diberikan oleh pembuat. Library tersebut dibuat oleh Aron Granberg dan versi yang akan digunakan adalah 3.8.1.

Pada desain data akan digunakan class diagram. Sedangkan pada desain tampilan akan terdapat beberapa desain berupa mockup dari permainan. Contoh

desain tampilan menu utama dapat dilihat pada gambar 1. Sedangkan untuk contoh desain tampilan edit custom tower dapat dilihat pada gambar 2.



**Gambar 1 Desain tampilan menu utama**

CustomTower name:

Custom Tower Upgrades

CustomTower Credit: 100 credit

Attack Speed Level:  
Not Upgraded yet

Attack Damage Level:  
Not Upgraded yet

Lower Tower Cost Level:  
Not Upgraded yet

Stun Chance Level:  
Not Upgraded yet

Instant Kill Level:  
Not Upgraded yet

Attack Range Level:  
Not Upgraded yet

Upgrade description

Confirm Changes

Lv1 Lv2 Lv3 Lv4 Lv5

Lv1 Lv2 Lv3 Lv4 Lv5

Lv1 Lv2 Lv3 Lv4

Lv1 Lv2 Lv3

Lv1 Lv2 Lv3

Lv1 Lv2

**Gambar 2 Desain tampilan editor custom tower**

## **5. IMPLEMENTASI**

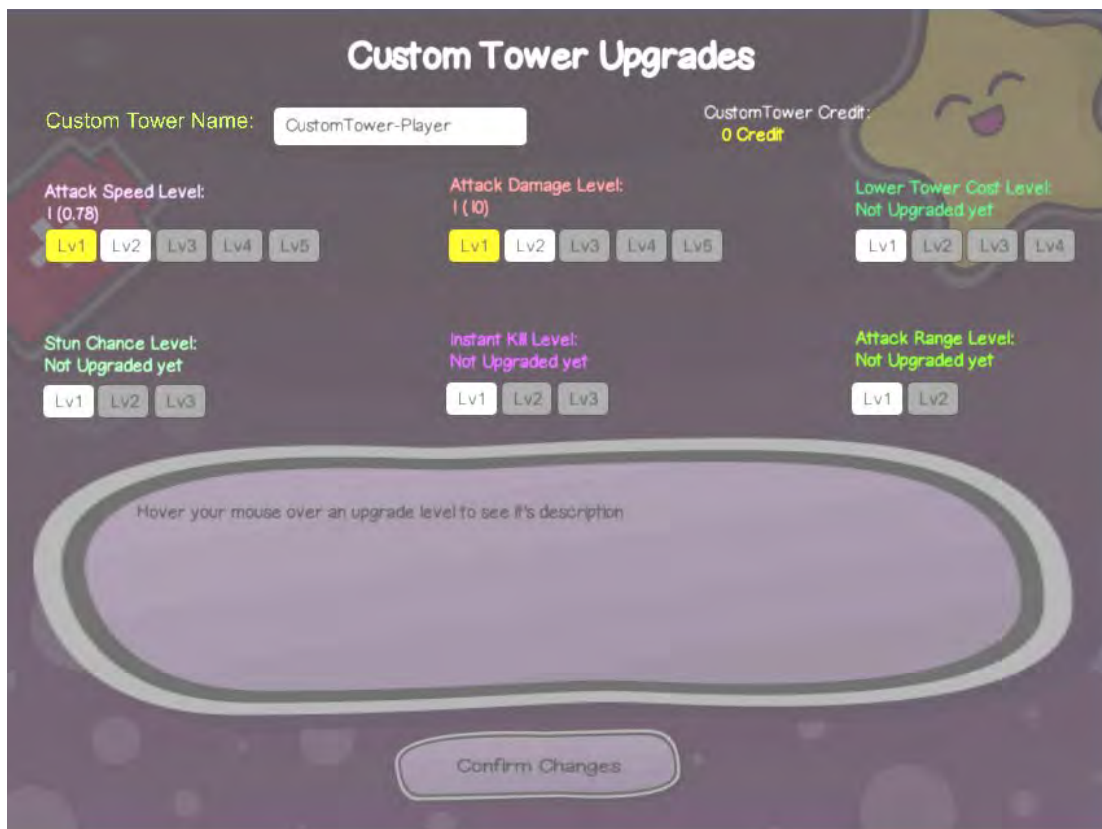
Implementasi yang dilakukan meliputi implementasi desain tampilan , implementasi class diagram dan proses dan implementasi pencarian jalan. Beberapa contoh dari implementasi tampilan yang dibuat dapat dilihat pada gambar 3 dan 4. Pada gambar 3 ditampilkan hasil implementasi dari menu utama. Tampilan menu utama dibuat menggunakan sistem canvas yang telah disediakan oleh Unity.



**Gambar 3 Tampilan Menu Utama**

Pada gambar 4 ditampilkan hasil implementasi dari edit custom tower. Menu edit custom tower berfungsi untuk mengubah atribut dari custom tower pemain. Tampilan pada bagian ini dibuat menggunakan fitur canvas dari Unity.

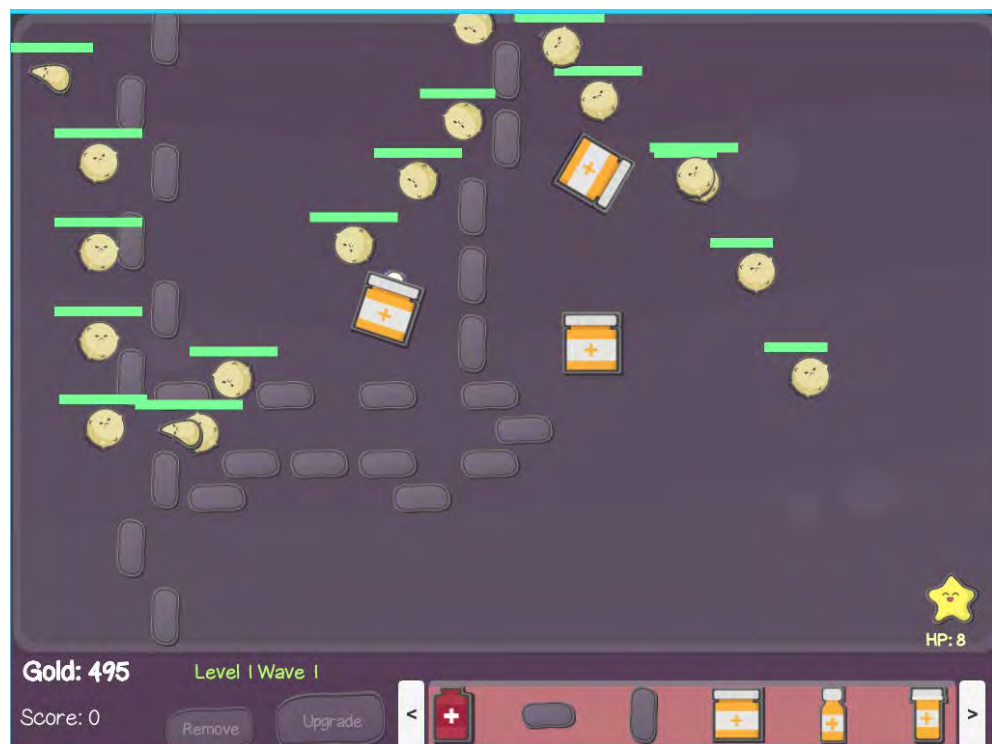
Pada bagian ini, user dapat melihat keterangan dari upgrade yang akan dipilih dengan cara melakukan hover pada button level dari suatu atribut.



**Gambar 4 Tampilan edit custom tower**

Pada saat permainan berjalan, user dapat membuat tower dan tembok, namun tidak dapat membuatnya diatas tower atau monster lain. Kemudian, monster akan mencari jalan terpendek menuju obyektif berupa bintang. Apabila user membuat tower atau tembok yang membuat jalan menuju obyektif menjadi terhalang, maka bangunan yang baru saja dibuat tersebut akan diberi tanda dan diabaikan. Saat monster melewati sebuah bangunan yang telah diberi tanda, maka bangunan tersebut akan dihancurkan. Tampilan saat permainan dimulai dapat dilihat pada gambar 5.





**Gambar 5 Permainan saat berjalan**

## **6. UJI COBA**

Terdapat 2 tahap dalam uji coba dan evaluasi ini yaitu verifikasi dan validasi. Tahap proses verifikasi dilakukan untuk mengetahui apakah permainan yang dibuat dapat berjalan dengan benar. Sedangkan tahap proses validasi dilakukan untuk mengetahui apakah permainan yang dibuat telah sesuai dengan yang diharapkan. Proses verifikasi dilakukan dengan pengujian program yang dibagi menjadi 4 bagian yaitu uji coba load custom track, uji coba custom tower, uji coba save tower, uji coba pencarian jalan dan uji coba pembuatan tower.

Proses validasi dilakukan dengan cara mengambil sampel 10 orang yang bersedia memainkan permainan ini. Responden yang akan diambil berasal dari mahasiswa Universitas Surabaya. Kemudian, peserta akan bermain dengan 2

mode permainan yang berbeda. Pada mode pertama, pemain diberikan kebebasan untuk membuat custom tracknya dan towernya sendiri. Sedangkan pada mode kedua, pemain hanya diberi kebebasan untuk membuat tower. Dari hasil kuisioner didapatkan hasil sebagai berikut:

- Permainan dengan custom track membuat permainan bergenre tower defense menjadi lebih seru untuk dimainkan.
- Permainan tower defense yang dibuat dirasa cukup mudah untuk dapat dimengerti.
- Panduan yang diberikan sudah cukup jelas.
- Jumlah tower dan wave sudah cukup menantang.

## **7. KESIMPULAN DAN SARAN**

Berdasarkan hasil uji coba yang telah dilakukan pada bab sebelumnya, maka dapat diambil beberapa kesimpulan permainan tower defense yang telah dibuat. Kesimpulan yang dapat ditarik adalah sebagai berikut :

- Fitur custom tower dan custom track yang ada dalam permainan tower defense ini meningkatkan tingkat keseruan pemain dalam memainkan game tower defense.

Saran yang dapat diberikan untuk pembaca yang ingin mengembangkan atau melanjutkan permainan ini adalah sebagai berikut :

- Menambah fitur multiplayer. Fitur ini memungkinkan permainan dimainkan oleh 2 orang yang saling bermusuhan. Sehingga pemenang ditentukan dari pemain yang objektifnya hancur terlebih dulu. Pemain

dapat mengirimkan monster dan custom monster sehingga varian monster dalam suatu wave dapat bertambah.

- Menambah jenis atau varian dari tower dan monster. Hal ini dapat dilakukan dengan menambahkan varian monster yang lebih kuat dan beragam untuk wave atau level selanjutnya.

## **DAFTAR PUSTAKA**

- Dalmau, D. S. (2004). Core techniques and algorithms in game programming. Diakses dari <https://books.google.co.id/books?id=IfDouSUqOUIC>
- Patel, A. (2014, May 26). Red Blob Games: Introduction to A\*. Diakses 27 November 2016, dari <http://www.redblobgames.com/pathfinding/a-star/introduction.html>